

demanda II

A prototype for Requirement Capture & Formalisation

Müller, D.

Im Rahmen des Europäischen Forschungsprojekts KARE - Knowledge Acquisition and sharing for Requirement Engineering (Esprit 28916) - wurde ein Ansatz für wissensbasiertes Anforderungsmanagement unter Einbeziehung des sich in Entwicklung befindlichen Standards für repräsentierende Systemdaten ISO 10303 - STEP AP233 entwickelt.

Ein Teilaspekt dieses Ansatzes unterstützt die systematische Transformation von natürlich sprachlichen Anforderungen in eine modell-basierte Repräsentation der Anforderungen.

In the European research project KARE – Knowledge Acquisition and sharing for Requirement Engineering (Esprit 28916) - an approach to knowledge-supported requirement engineering and the relation to the emerging standard for representing system engineering data ISO 10303 - STEP AP233 has been developed.

A section of this approach features the systematic transformation from natural language requirements into a model-based representation of requirements.

1 Problem area

Natural language representation is the initial form of defining a requirement statement. Nowadays, the requirements for a product specification are mainly captured in text-documents of paper or electronic format. These documents establish the basis for the exchange of information between customer and supplier. They should include every kind of information linked with a product.

The extensive specification of requirements in natural language text is one of the main reasons for ambiguity and inconsistencies. The European research project KARE - Knowledge Acquisition and sharing for Requirement Engineering (Esprit 28916) intends to promote the current acquisition practices from a mainly document based approach to a system engineering model-based approach that allows the electronic interchange of all necessary data in a distributed engineering environment. The approach aims at producing consistent and complete product specifications in shorter times at substantially lower cost and with improved quality.

2 Requirement Formalisation

The systematic transformation of natural language requirement statements into a model-based representation is named **requirement formalisation**. Within this formalisation process, every given requirement can be evaluated step by step with human interaction. This will be supported by the requirement formalisation tool *demanda II*, developed within the context of KARE by the TU Clausthal, Institute for Mechanical Engineering (IMW).

The information, included in each requirement statement that is specified in natural text, is transformed into a system model description that represents the same information. This will be realised by identifying elements of a system specification equivalent to ISO 10303 - STEP AP233. These are first of all the physical architecture, functions, functional behaviour and properties. The content of the resulting model is of course not more correct or complete in its quality than the natural language requirement. However, the model improves the possibility to check automatically on inconsistencies, missing or incorrect requirements.

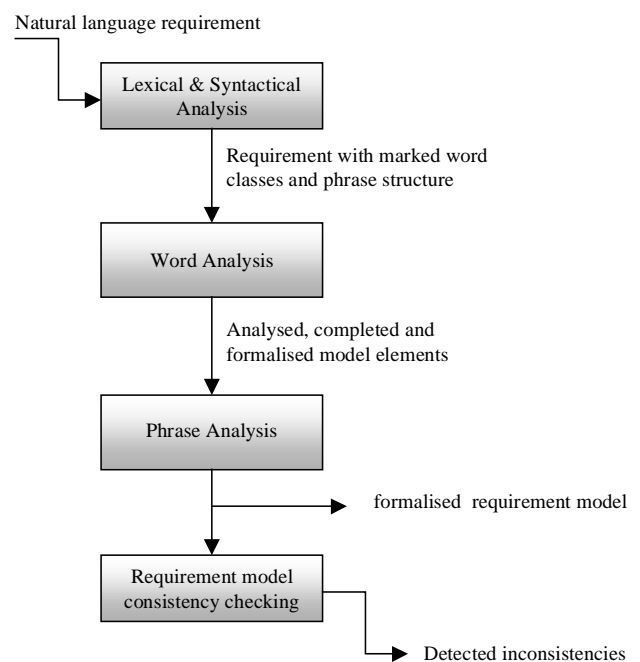


Figure 1: Sequences of the requirement formalisation

The developed formalisation process of /3/ defines the implementation of how to transform natural language requirements into a model-based representation. Within this systematic process, one will obtain a model-based representation step by step out of individual fragments. At the end of the process, there will be a rigorous formalised model that represents the requirements. The requirement formalisation process as defined in /4/ consists of three main phases. **Figure 1** displays an overview of sequences of the requirement formalisation.

The **lexical and syntactical analysis** converts a sequence of characters into an ordered set of symbols and identifies different parts of speech. In the lexical analysis, the sentence is split into its single components (words) and these words are classified into different word classes. The syntactical analysis (or parsing) provides the hierarchical phrase structure of the sentence. As natural language permits ambiguity in respect to the phrase structure of a sentence, the user may have to resolve the ambiguity manually.

Based on the marked word classes and the phrase structure of a sentence, the **word analysis** follows as second step. It identifies the meaning of the different symbols in respect to the STEP AP233 data model. In other words, symbols describing a conceptual component of a system like i.e. function or property are analysed, completed and formalised. The contemplation of the meaning takes place on the level of words.

As last step, the **phrase analysis** captures the relationship between the identified symbols. Possible relationships are the hierarchy of components, interfaces between components and other relationships between components, functions and properties.

The entire realisation of the formalisation process maps the information contained in the defined requirements onto the data model. At the end, there exists a rigorous formalised requirement model. As mentioned before, the content of the resulting model is of course not more correct or complete in its quality than the natural language requirement. But this model can be checked systematically on rules of consistency or completeness. The integrated knowledge tool as part of the KARE *Workbench*, as described in /2/, will cover this process. But to gain rational support, often the basic knowledge of a domain expert is needed for particular instances.

3 The Systems Engineering standard ISO 10303-AP233

The ISO 10303 - STEP AP233 specifies an application protocol (AP) for the systems engineering design data. The AP233 defines the context, scope and information requirements for various development stages during the design of a system. This application protocol shall be applicable to any form of systems, such as aircraft, cars, marine and plant.

These systems are made up of a set of predominantly active components, sensors, displays and actuators that are interconnected via dedicated direct links or by communication means. Systems cannot be simply considered as a sum of the single components; rather, their integrated behaviour and properties which in most cases are real-time dependent, has to be defined, validated and verified.

The systems engineering process models use techniques such as functional decomposition and abstraction as the means of tackling the complexity of the systems being developed.

The following domains are covered by the 5th Working Draft of the AP233 (see also /1/):

Requirements:

- Elicitation and analysis
- Tests definition

Functional Design:

- Data description support
- Function description support
- Behaviour description support
- OO Design

Physical Design:

- Topological architecture definition
- Function component mapping support

Graphical representation and layout

Traceability & Configuration management

Industry process:

- Risk management
- Justification management
- Documentation support
- Link with workflow definition
- Approval support
- Person allocation
- Date allocation
- Effectiveness management

It is very likely that all these domains will be included in the final AP233. A Publicly Available Specification for the Systems Engineering Data Representation will be available soon (PAS 20542).

4 An overview of the Requirement Capture and Formalisation Tool *demanda II*

The KARE *Workbench* is the prototype demonstrator for the profit of integration of a model-based approach into the early development phase of a complex product within a distributed environment.

Beside a Requirement Management Tool and a Knowledge Management Tool, one integrated part of the KARE *Workbench* is the Requirement Capture and Formalisation Tool *demanda II*. This prototype is a completely new development by the IMW. It supports the user to capture, formalise and manage natural language requirements on the level of a data model with lots of supplied functionality. **Figure 2** specifies a general overview of the supported functionality.

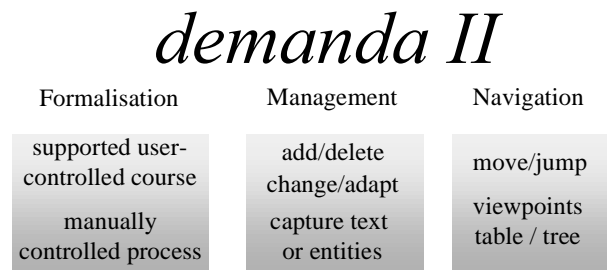


Figure 2: Supported functionality of *demanda II*

Formalisation – One main task of *demanda II* is the user-support to formalise natural language requirements. There are two possible courses to perform the transformation of specified system requirements into a model-based representation.

One opportunity is to be led systematically through the formalisation process. A user-controlled course guides step by step to a model-based representation of the corresponding requirement. This process is not fully automated, because of the complexity of speech. So the user still has the possibility to influence the input to the information repository.

A sophisticated user will have the possibility to abstain from the given procedure to realise a formalisation manually. The general offered functionality of *demanda II* to manage and create entities enables the user to realise a complete or partial mapping of the formalisation based on his own knowledge and interpretation.

Management – *demanda II* volunteers the user with many management facilities. The file exchange is currently based on STEP AP233 Part21 files (Step physical files). The exchange interface of *demanda II* covers the complete scope of AP233.

Processed files of a Requirement Management Tool establish the basis for *demanda II*. For more or less unformalised files, the complete scope of management facilities are applicable to realise and define a formalisation. For files that have already passed a formalisation procedure, the normal functional range is at a user's disposal to handle, adapt or change these represented requirements. In general, the user can create, delete instances or he can change, add or adapt the content of single attributes of an existing data model instance.

Altogether the tool offers a complete functionality to handle the content of the data model. And because the concept of *demanda II* is based on the level of the data model, the basic functionality hands a wide scope of freedom to act on and with the data model, not restricted by incomplete interfaces.

Navigation – For a better overview and control of the content of the data repository, *demanda II* offers the possibility to obtain information and navigate directly on the level of the data model.

This means explicitly, different view facilities enable the user to get information of the actual content of the data model repository, alternatively on the level of entities or instances. At the beginning this seems to be a little bit complicated, but with some experiences with the data model the user will appreciate this functionality. This offers the opportunity to get complete control over the content of the model and the represented information. This saves time to receive a correct and consistent model representation.

5 The GUI of *demanda II*

The architecture of *demanda II* consists of three assembled components. The prototype is based upon a runtime module, generated by the ECCO Toolkit - Version 2.3.0 from PDTec GmbH, Germany. For the compilation of the runtime module, the EXPRESS source code of ISO 10303 - STEP AP233 WD5.2 has been used as underlying data model repository. The GUI of *demanda II* puts on these two components. It has been implemented within the tool command language Tcl/Tk (Version TclPro 1.4.1 - bundles Tcl/Tk 8.3.2 and compatible versions of TclX and [incr Tcl]) by the IMW. **Figure 3** offers an overview of the GUI.

The main window of the GUI is structured into three parts. Below the menu bar and the used schema information on top of the window, the user will find the text capture and operating window. This is the

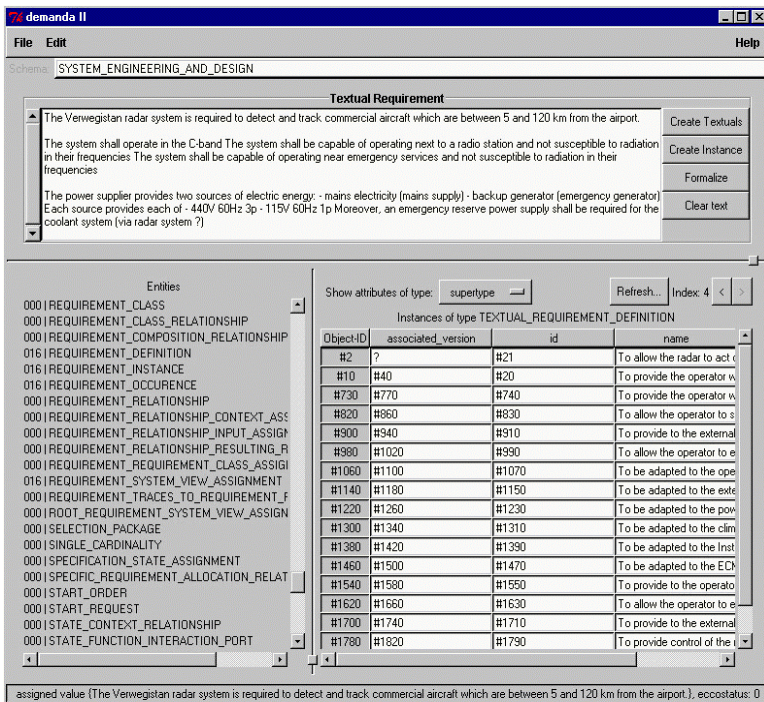


Figure 3: Overview of the GUI of *demanda II*

main window for capturing new textual requirements and to start with different parts of formalisation. The text capture window owns the basic functionality of a text editor. This entered text represents the starting point for a formalisation. The row of buttons on the right side will initiate different operations of formalisation. There exists the opportunity to create a new textual requirement, a new instance of any desired data model entity or start a supported requirement formalisation.

The lower part of the main window serves the management and navigation facilities. On the left side of the window the complete list of AP233 entities is displayed in alphabetical order. To get extended information about the instances, selection with the mouse will display the content of all related instances of an entity on the right side. The content will be displayed in a table in linear order of the object-id's. Left mouse click on an attribute will display an interactive window with the content of the chosen instance. The values can be changed, added or deleted.

6 Conclusions

demanda II is a complete new development and therefore a first but working prototype. It is already a powerful tool that offers some new functionality to act on a systems engineering data representation. The architecture offers a great variability in terms of adaptation and updating. With the exchange of new modules, it will be possible to keep it up-to-date with less effort. Because of the complexity of

speech and limited time of development, the automation of the requirement formalisation is not sophisticated. But considering that the technical basis is already existing, with the big advantage of an complete supported interface, the tool represents the basis for further developments. These developments will look into a predominant automated requirement formalisation, an enhanced GUI with additional functionality for a more user-friendly environment for not so sophisticated users and an extension of the interface for XML support to enable web based technologies.

The KARE project - Knowledge Acquisition and sharing for Requirement Engineering - is co-funded by the European Commission under the ESPRIT programme, No. 28916. The author wishes to acknowledge the Commission for their

support. He furthermore wishes to acknowledge the KARE project partners for their contribution during the development of various ideas and concepts. For further information please check out the KARE web-pages <http://www.kare.org/>.

7 Summary

demanda II is a prototype for requirement capture and formalisation. The requirement formalisation features the systematic transformation of natural language requirements into a model-based representation. As referenced data model is used the ISO 10303 – STEP AP233 for systems engineering data representation. Embedded into the KARE Workbench architecture and concept, *demanda II* supports the process of knowledge supported requirements engineering on supplier and acquirer side.

8 Literature

- /1/ N.N.: <http://www.sedres.com/>, 2001
- /2/ K. Heimannsfeld, D. Müller: Requirements Engineering Knowledge Management based on STEP AP233, Institutsmitteilung Nr. 25, Institut für Maschinenwesen, TUC 2001
- /3/ Heimannsfeld, K.: Modellbasierte Anforderungen in der Produkt- und Systementwicklung; Shaker Verlag, Aachen 2001
- /4/ D. Müller, K. Heimannsfeld: The KARE Formalisation Process Specification and Model Usage Guide, KARE Internal paper, 2001